

Mastra

Open-source agent framework for Typescript

1. Create an Agent

```
import { Agent } from "@mastra/core/agent";
import { openai } from "@ai-sdk/openai";

export const myAgent = new Agent({
  name: "My Agent",
  instructions: "You are a helpful assistant.",
  model: openai("gpt-4o-mini"),
});
```

Mastra

Open-source agent framework for Typescript

2. Structured Output

```
const schema = {
  type: "object",
  properties: {
    summary: { type: "string" },
    keywords: { type: "array", items: { type: "string" } },
  },
  additionalProperties: false,
  required: ["summary", "keywords"],
};

const response = await myAgent.generate(
  [
    {
      role: "user",
      content:
        "Please provide a summary and keywords for the following text: ...",
    },
  ],
  {
    output: schema,
  },
);

console.log("Structured Output:", response.object);
```

Mastra

Open-source agent framework for Typescript

3. Creating tools

```
import { createTool } from "@mastra/core/tools";
import { z } from "zod";

const getWeatherInfo = async (city: string) => {
  // Replace with an actual API call to a weather service
  const data = await fetch(`https://api.example.com/weather?city=${city}`).then(
    (r) => r.json(),
  );
  return data;
};

export const weatherInfo = createTool({
  id: "Get Weather Information",
  inputSchema: z.object({
    city: z.string(),
  }),
  description: `Fetches the current weather information for a given city`,
  execute: async ({ context: { city } }) => {
    console.log("Using tool to fetch weather information for", city);
    return await getWeatherInfo(city);
  },
});
```